

Topographic Visualization of Prefix Propagation in the Internet

Pier Francesco Cortese, Giuseppe Di Battista, Antonello Moneta, Maurizio Patrignani, and Maurizio Pizzonia

Abstract— We propose a new metaphor for the visualization of prefixes propagation in the Internet. Such a metaphor is based on the concept of topographic map and allows to put in evidence the relative importance of the Internet Service Providers (ISPs) involved in the routing of the prefix. Based on the new metaphor we propose an algorithm for computing layouts and experiment with such algorithm on a test suite taken from the real Internet. The paper extends the visualization approach of the BGPlay service, which is an Internet routing monitoring tool widely used by ISP operators.

Index Terms—Interdomain Routing, Internet Visualization, Graph Drawing, Spring Embedder

1 Introduction

The Internet can be visualized at different abstraction levels for different purposes. For example, if one is interested in all the technical features of the network, then the Internet can be visualized with an high level of detail, showing the local area networks, the routers, and the point-to-point links. On the other extreme, if one is interested only in the relationships between Internet Service Providers (ISPs), then the Internet can be visualized at the Autonomous Systems (AS) level, where an AS is a collection of networks under the same administrative authority. At such a level, the Internet is currently partitioned into more than 20,000 ASes. An on-line survey about the visualization of the Internet can be found in [1].

Many research efforts have been done for visualizing the Internet at the AS level. Some systems aim at visualizing the entire Internet structure (see, e.g., [24, 7, 17, 2, 3]), while others visualize a small portion of it with different specific purposes. For example [14] describes a system that visualizes the relationships between ASes that are stored into an Internet Registry. The BGPlay system [16] displays the evolution over time of the routing paths traversed by the traffic to a selected AS. BGPlay is a quite popular tool between ISPs operators and has been adopted by international organizations that publish real time collected data about inter-AS routing [5, 6]. A project that has a similar goal is presented in [25].

However, the research on the visualization of the Internet has to face new interesting challenges. In fact, recent studies [23, 10] have shown that the ISPs (and then the ASes) of the Internet are self-organized into a customer-provider hierarchy (in the following *Internet Hierarchy*). In this hierarchy, customers buy from their providers a transit service for their traffic to and from the Internet. In [30] each ISP is assigned to a level, according to its rank in the Internet Hierarchy. At this point, the problem of visually representing the Internet Hierarchy naturally arises.

An amazing number of methods have been devised to visually represent hierarchies in a variety of application domains (see, to give a few examples, [9, 37, 12, 13]) many of which are targeted to show very large hierarchies.

In this paper we address the problem of the simultaneous visualization of the Internet Hierarchy and of other Internet features that are interesting from some applicative perspective. More specifically, we extend the visualization paradigm that has been successfully adopted in BGPlay [16] enriching it with support for the Internet Hierarchy vi-

ualization. Fig. 1 shows a screenshot of the BGPlay system. The red vertex (AS137), indicated by an arrow, is the AS we are focusing on. The picture represents paths to reach AS137 from several other ASes at a certain time. For example, the path AS16150, AS6939, AS6762, AS137 is shown to be used for traffic incoming AS137 from AS16150. No information is available in this visualization about the relevance of ASes and their economic relationships.

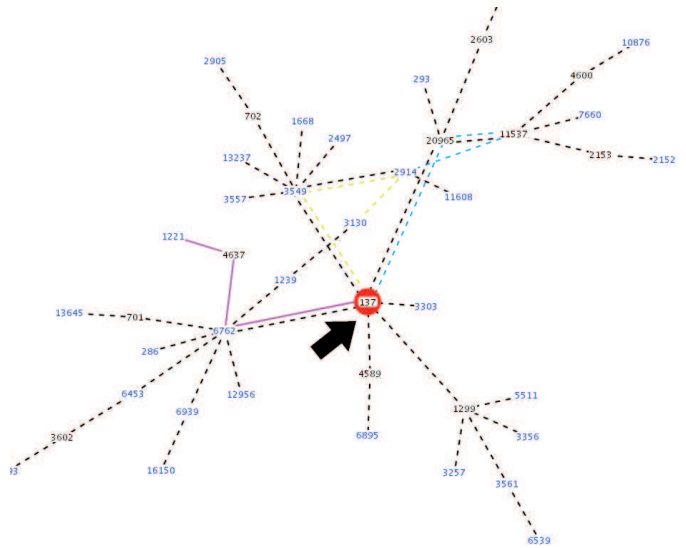


Fig. 1. A screenshot of the BGPlay system showing the same AS-graph shown in Fig. 2(a).

Fig. 2(a) shows a screenshot of BGPlay enhanced with the visualization system presented in this paper. To visualize the Internet Hierarchy we use the metaphor of a topographic map. The contour lines are used to confine ASes that are at the same level of the hierarchy. For example, the ASes inside the central brown area are top level ASes. As the “hill” decreases in height the ASes decrease their rank in the hierarchy. The map shows quite well that some of the traffic flows have to climb the entire hierarchy to reach AS137, while other flows take “shortcuts”. For example, the path AS16150, AS6939, AS6762, AS137 does not pass through the Internet backbone since AS6939 and AS6762 exchange traffic at a lower level. Such kind of paths are usually more efficient and less expensive than paths that pass through the Internet backbone. An example of a real topographic map is show in Fig. 2(b); more detail about topographic maps can be found in [8].

Observe that drawing a network within the above described metaphor can be interpreted as a special case of a clustered graph drawing problem (first introduced in [20]), where the hierarchy describes a simple structure of clusters. Several authors dealt with the problem of representing a clustered graph using a spring embedder ap-

• P. F. Cortese, G. Di Battista, A. Moneta, M. Patrignani, and M. Pizzonia are with the Roma Tre University, Dipartimento di Informatica e Automazione, Rome, Italy. E-mail: cortese@dia.uniroma3.it, gdb@dia.uniroma3.it, antonello.monetsen@gmail.com, patrignani@dia.uniroma3.it, pizzonia@dia.uniroma3.it.

Manuscript received 31 March 2006; accepted 1 August 2006; posted online 6 November 2006.

For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org.

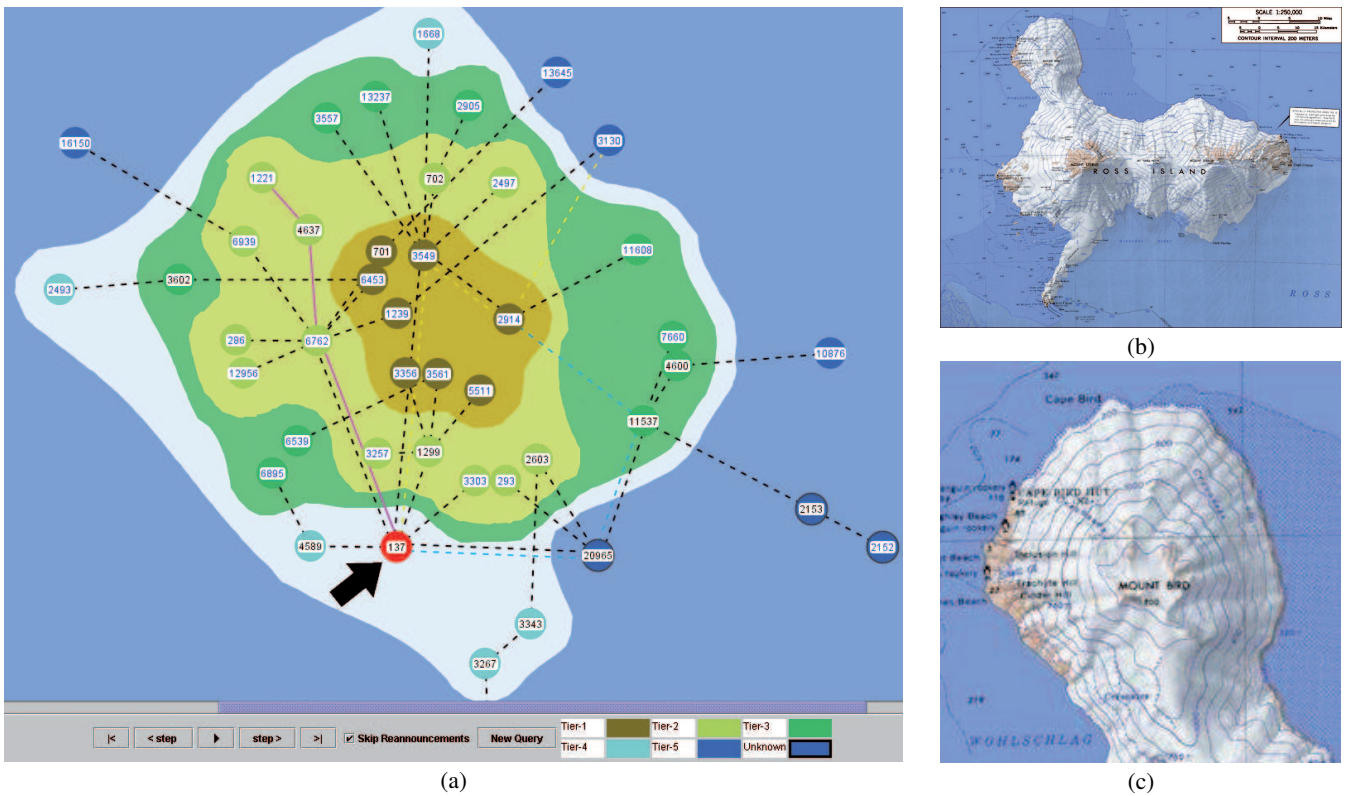


Fig. 2. (a) A screenshot of the BGPlay system enhanced by the topographic map approach described in this paper. The AS that originates the prefix is 137, highlighted in red and indicated by an arrow. Our approach clearly shows the importance of the ASes traversed by paths ending into 137. (b), (c) An example of a real topographic map (courtesy of the U.S. GS).

proach. Eades and Huang [19] proposed a system for the visualization of huge graphs, by first performing a clustering and then visualizing a portion of the graph by applying a force-directed approach. Walshaw [36] introduced a heuristic method for drawing large graphs which uses a multilevel technique combined with a force directed placement algorithm. Frishman and Tal [21] proposed an algorithm for dynamic drawings of clustered graphs.

In this paper we present the following results.

- We propose (Section 3) a new metaphor for showing the Internet Hierarchy, the interconnections among the ASes and the routing at the AS level from specific observation points. The metaphor is specifically tailored to be used within the BGPlay service.
- We show a visualization algorithm that computes topographic maps of the observed portion of the Internet Hierarchy (Section 4).
- We describe an implementation of the algorithm into the BGPlay system and analyze its effectiveness with a wide experimental analysis (Section 5).

The paper contains also background terminology (Section 2) and concluding remarks (Section 6).

2 Background

We provide some networking background and a brief description of the BGPlay routing visualization system.

2.1 Networking

In the Internet each host is identified by an *IP address*. An *IP prefix* identifies a set of (contiguous) IP addresses having the same leftmost n bits, (e.g. 193.204.0.0/15 indicates a prefix 15 bits long) [34]. Similarly to telephone call routing, inter-domain routing in the Internet is

based on the destination prefix. Since a prefix identifies a set of addresses, it implicitly identifies a set of hosts having such addresses.

An *Autonomous System (AS)* is a portion of the Internet under a single administrative authority. An Internet Service Provider (ISP) typically runs one or more ASes. Each AS is identified by an integer number. Traffic starting from an AS and directed to a specific prefix traverses a sequence of ASes called *AS-path*. The configuration of such paths on the routing devices is too complicated to be manually performed. Hence, ASes exchange routing information with other ASes by means of a routing protocol called *Border Gateway Protocol (BGP)* [29]. BGP is based on a distributed architecture where *border routers* that belong to distinct ASes exchange the information they know about the reachability of prefixes. Two border routers that directly exchange information are said to have a *peering session* between them, and the ASes they belong to are *adjacent*.

The *AS-graph* is the graph having one vertex for each AS and one edge between each pair of adjacent ASes. Note that the AS-graph is not a multi-graph.

Each router stores information about routing in its *Routing Information Base (RIB)*. The RIB is a table where each row is a pair (prefix, AS-path) meaning that a certain prefix is reachable through the associated AS-path. Such pairs are called *routes*. The main purpose of BGP is to allow the routers to exchange the routes they know. Since RIBs may be huge, the BGP process running on a router sends to its peers the full RIB only when a peering session is set up. During regular operation only changes to the RIB, called *updates*, are sent.

Routes related to a certain prefix begin their existence within an AS called the *originator* of the prefix (typically the AS to which the prefix belongs). These routes are propagated by means of updates to adjacent ASes, which in turn propagate it to adjacent ASes, etc. Every time a router propagates an update, it prepends its AS identifier to the AS-path; thus, the AS-path of an update is the list of ASes that the update has passed through.

2.2 The Internet Hierarchy

ASes cooperate in order to ensure good connectivity service to their customers but are competitors from a commercial point of view. Commercial relationships among ISPs directly influence how BGP routers are configured. Router configurations affect, in turn, the AS-paths that can be announced in the Internet. A valid AS-path is usually made of three consecutive parts [30]: (i) it first traverses the Internet walking on the edges of the AS-graph in the direction from customers to providers, (ii) it optionally traverses one peer-to-peer relationship between two ASes of the same relevance, and (iii) it traverses the Internet walking on the edges of the AS-graph in the direction from providers to customers till the destination is reached.

In an ideal world, *tier 1 ASes* do not have any provider and are involved in peer-to-peer relationships with all the other tier 1 ASes; further, *tier- i* ASes are customers of ASes of tiers greater than i , are involved in peer-to-peer relationships with other tier- i ASes, and are providers of ASes of tiers less than i . From BGP routing tables it is possible to infer, with a few errors, the tier of the each AS [30].

2.3 The BGPlay Internet Visualization Service

BGPlay is a service that displays the portion of the AS graph that describes how the traffic flows to a certain AS from a set of selected ASes. It obtains routing data from well-known and publicly available sources, namely the Routing Information Service (RIPE NCC) [5] and the Route Views project (University of Oregon) [6] whose archives are used for network debugging purposes or scientific investigation and are updated in real-time.

To query BGPlay, the user connects to a Web page, which hosts BGPlay, and starts the BGPlay applet which presents a query window asking for a prefix and a time interval. When the user submits the query, BGPlay processes the request and displays the animation window (Fig. 1), which presents the routing information.

The main part of the window shows an AS-graph and the routing at a certain instant of time for the prefix specified by the user. Each number represents an AS, and the AS originating the prefix is placed in the center of the graph and highlighted by a red circle. Each solid or dashed line represents a segment of an AS-path seen by RIS or Oregon Route Views. An AS-path starts in the originating AS and stops in an AS which provides BGP routing data to RIS or Oregon Route Views. The AS-paths that do not change during the query interval are merged into trees rooted at the origin AS and drawn dashed. Each tree is drawn in different color so that it can be unambiguously identified. AS-paths that change during the query interval are drawn solid and are not merged.

BGPlay can animate the evolution of the routing over time. A panel, on the left of the window, shows the density of routing events over the interval of time of the query and highlights the instant of time whose routing is currently shown in the main part of the window.

3 Choosing the Visualization Metaphor

When we had to select a visualization metaphor for showing at the same time both the AS-paths of the BGP routing and the levels of the ASes hierarchy, we considered several possible options and focused on three main choices.

1. The classical approach of Sugiyama *et al.* [33] visualizes a hierarchy by assigning the y -coordinate of each vertex (AS) according to its rank. In this way, top (bottom) level vertices appear in the upper (lower) part of the drawing.
2. Another typically adopted strategy for visualizing a hierarchy [15] was to represent ASes with a size and/or color proportional to their rank in the hierarchy.
3. A further possibility was to extend the visualization approach already used for BGPlay [16] by placing the ASes within a “topographic map” where each AS falls into the region assigned to its level.

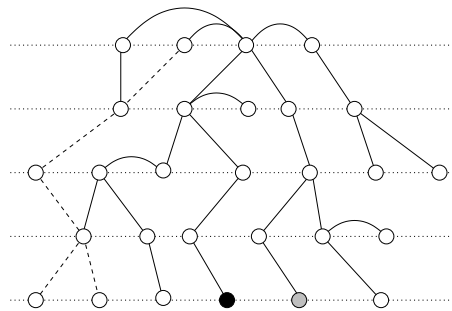


Fig. 3. An handmade layered drawing showing the drawbacks of this kind of representation. Observe how paths zigzag while climbing and descending the hierarchy. Also, ASes that are very far in the network may be represented by nodes which are very near in the drawing (as the black and the gray nodes).

We deliberately ruled out three-dimensional representations, since, apart from the fascination of such a scenario, effectively browsing a three-dimensional environment involves the use of sophisticated interfaces, while network operators and researchers appear to be reluctant to adopt particularly complex software architectures and tools.

Before discarding options 1 and 2 we performed several experiments against typical routing scenarios obtained by querying the Routing Information Service of the RIPE NCC and equipping this information with the Internet hierarchy computed with the algorithm of [30]. The purpose of the experiments was to compare the readability of the three graphic metaphors above from an information visualization perspective. To this aim we drew several diagrams according to the three strategies.

From our experiments we could observe the following pros and cons for the three metaphors

The traditional hierarchical representation through layered drawings, although very effective in conveying the hierarchy information associated with the ASes, turned out to be unsatisfactory for our application. In layered drawings, in fact, the x -coordinates of the nodes are generally chosen in order to reduce edge crossings. It is often the case that a climbing path zigzags horizontally the diagram (see Fig. 3), reducing its readability. Also, two nodes which are one near the other on the same level (see the black and the gray nodes of Fig. 3) may represent ASes that are actually very distant in the network. On the other hand, the number of edge crossings jumps up when the x -coordinates of the nodes are somehow constrained. Finally, many of the algorithms used for producing hierarchical representations do not allow links between nodes of the same layer, which are very common in practice. Some examples of layered drawings with intra-rank edges can be found in [28, 35].

Using size and color to suggest the rank of the AS in the hierarchy has the advantage of being a very simple strategy, easy to implement and to plug into existing visualization systems, but, on the other hand has the major disadvantage of depriving the user of an overall view on the ASes of the same level. ASes are scattered in the drawing, and there is no clue of how many levels of the hierarchy are involved and how many links span more than one level.

The “topographic map” strategy produced the most clear and intuitive drawings. In such metaphor, ASes of the same rank are drawn inside the same region and regions corresponding to a higher level are contained inside the regions corresponding to the lower ones. Further, this metaphor is consistent with well-known drawing conventions in the area of inter-domain routing where the AS-paths are merged into a graph, which is colored in such a way to make it possible to uniquely reconstruct the path from each AS to the AS advertising the prefix.

More in detail, in our approach, the tiers of the ASes play the role of the elevation. We can imagine the drawing as a topographic map in which ASes of tier 1, the top level, are drawn in the middle, resembling the peak of a mountain, and ASes of lower levels are drawn progressively farther, as if they were on the downhill of the mountain.

More formally, each tier i is associated with a connected region of the plane that we denote A_i with $i = 1 \dots T$, where T is the number of tiers. Region A_i , with $i = 2, \dots, T$, has a hole which contains region A_{i-1} . All and only ASes of tier i are contained into region A_i . The shape of the regions is a critical issue, since it constrains the layout of the ASes and hence may reduce the effectiveness of a layout algorithm in obtaining drawings with small edge length, even ASes distribution, and small number of crossings. In an ideal topographic map the regions have not necessarily a regular shape, but smoothly surround the ASes that they contain. Region shapes and ASes positions are strictly related, since in order to be enclosed into a region the ASes of the same tier should be near one to the other.

We call *edge region crossings* of a map the number of edges that cross the border of the same region more than once. Consider that when an edge connects two nodes in different regions a crossing between such edge and a region border occurs. On the other hand, each edge should cross the border of a region at most once. For readability reasons, we prefer topographic maps with no edge region crossing.

The algorithm described in Section 4 produces such drawings which look very much like a topographic map with a superimposed AS-graph.

Concerning the representation of the AS-paths and of the way they change over time, we adopt the same drawing standard adopted by BGPlay and described in Section 2.3. An example of the final result is shown in Fig. 2.

4 Layout Algorithm

In this section we describe the layout algorithm for producing drawings according to the visualization metaphor described in Section 3.

We adopt a force directed approach, also called *spring embedder*, which computes the final layout of the input graph as the equilibrium configuration of a simulated physical system where the nodes are represented by mutually repulsing electric charges and the edges by springs with a given stiffness and a natural length [22, 18]. Several factors encourage such a choice: (i) Since its physical system can be enriched with additional forces, a spring embedder is easily tailorable to account for additional constraints on the drawings [31, 32, 18]. (ii) Spring embedders have already been used to produce drawings of clustered graphs in which the nodes of each cluster are constrained to stay inside the same region of the plane [19]. (iii) The original BGPlay interface obtains pleasant and effective results with a spring embedder algorithm.

Our visualization metaphor is quite different from the drawing conventions for which spring embedders are usually applied. In fact, our regions are all contained one inside the other and their boundaries have not a fixed shape. Therefore, we devised a new model whose main ingredients are the following.

- In order to enclose one region into the other, and force the ASes to belong to their assigned region, we introduce one border region B_i at a time, starting from the inner region B_1 and ending with B_{T-1} , where T is the number of tiers (ASes of tier T have no boundary). When a border is introduced, additional forces are added to the physical model to ensure that the ASes are enclosed into the border of their region. In this phase the region borders are constrained to be circular. The incremental introduction of the borders will make edge region crossings unlikely to occur.
- In order to allow borders B_i to change their shape and adapt to the position of the ASes (as required by the topographic metaphor described in Section 3), we represent each border with a chain of very close special nodes, called *border nodes*. When all region borders are introduced, these border nodes are allowed to move. Also, by suitably removing or adding border nodes to the chains, we control their density, since a loose chain of border nodes would easily be traversed by an AS, while a too dense chain would wind on itself.

A detailed description of our layout algorithm follows. The input of the algorithm is the AS-graph where each AS is labeled with a tier

number in $\{1, \dots, T\}$. The output of the algorithm is a layout of the AS-graph and the borders that separates areas A_1, \dots, A_T where each area contains all ASes associated with its tier as stated in Section 3.

The layout algorithm is composed of two phases.

Phase 1. A set of $T - 1$ concentric circular borders B_1, \dots, B_{T-1} are identified on the plane, with radius $r_1 < r_2 < \dots < r_{T-1}$. We denote by A_1 the area enclosed by B_1 and by A_i , with $i = 2, \dots, T - 1$, the area enclosed by B_i and outside B_{i-1} . The purpose of this phase is to ensure that, for each tier i , all ASes belonging to tier i are placed inside area A_i . This is obtained by means of an incremental layout strategy that starts from ASes of Tier 1 (see Fig. 4).

Consider, for each boundary B_i of radius r_i , circles B_i^- and B_i^+ with radius $r_i^- = r_i - \varepsilon$ and $r_i^+ = r_i + \varepsilon$ respectively, where ε is a very small number with respect to the radius of B_i .

1. ASes of Tier 1 are confined inside B_1^- by a force f_1^- that attracts these ASes toward the center of the drawing and that is null outside B_1^- . All other ASes are kept outside B_1^+ by a force f_1^+ that repulses these ASes from the center of the drawing and that is null inside B_1^+ . With these additional forces the spring embedder is run until an equilibrium is reached.
2. The space between B_1^- and B_1^+ is used to place a chain of border nodes called *fence*. Border nodes cannot move in this phase but they repel ASes and curb them inside their region. With B_1 represented by border nodes the spring embedder is run until equilibrium is reached. See Fig. 4(a).
3. Operations of steps 1 and 2 are iteratively applied for Tiers $2, \dots, T - 1$. That is, for each Tier i , the spring embedder is run with new forces, f_i^- and f_i^+ , to contain inside B_i^- vertices of tiers less than or equal to i and outside B_i^+ vertices of tiers greater than i . A new fence is added and a new spring embedder run is performed. Throughout the computation, the forces introduced in the previous steps are not dropped, but remain to enforce proper confinement of each AS within its area. See Figs. 4(b)-(c).

Phase 2. Forces f_i^- and f_i^+ , for $i = 1, \dots, T - 1$, are removed, and the border nodes are left free to move until equilibrium is reached. In order to keep the border density within a safe range, edges of the fences that are longer than a certain threshold are split and edges that are shorter than a certain threshold are contracted. Also, the force exerted on a border node is multiplied by a *mobility factor* that allows to fine-tune the rigidity of the fences. An example of the result of Phase 2 is shown in Fig. 4(d), while the corresponding picture presented to the user is shown in Fig. 2(a).

The algorithm described above has been prototypically implemented in the BGPlay system. No change has been made to the BGPlay architecture described in [16]. The information about levels of the ASes is computed with the techniques shown in [30]. Our experiments have been performed on a PC with an AMD AthlonXP 2600 processor – 2.133 GHz and with 0.5 GB of RAM. For our test suite running times range from 7 to 62 seconds with an average 27.83 seconds.

5 Experimental Evaluation

In order to experimentally evaluate the effectiveness of the proposed algorithm we need to consider both traditional graph drawing aesthetic criteria and aesthetic criteria suitable for the specific application and the adopted visualization metaphor.

Concerning the first ones, in the graph drawing and visualization literature several measures have been proposed to evaluate the aesthetic quality of the straight-line drawings obtained with force directed methods (see, e.g., [11]). We decided to adopt the following measures. **Normalized number of edge crossings (EEX)**, i.e. the ratio between the number of edge crossings and the total number of edges, which is generally considered the most prominent readability measure [26, 27]. **Normalized edge-length standard deviation**, i.e., the ratio between the standard deviation and the average length of the edges (ELSD). In fact, a readable drawing should have an homogeneous length for its

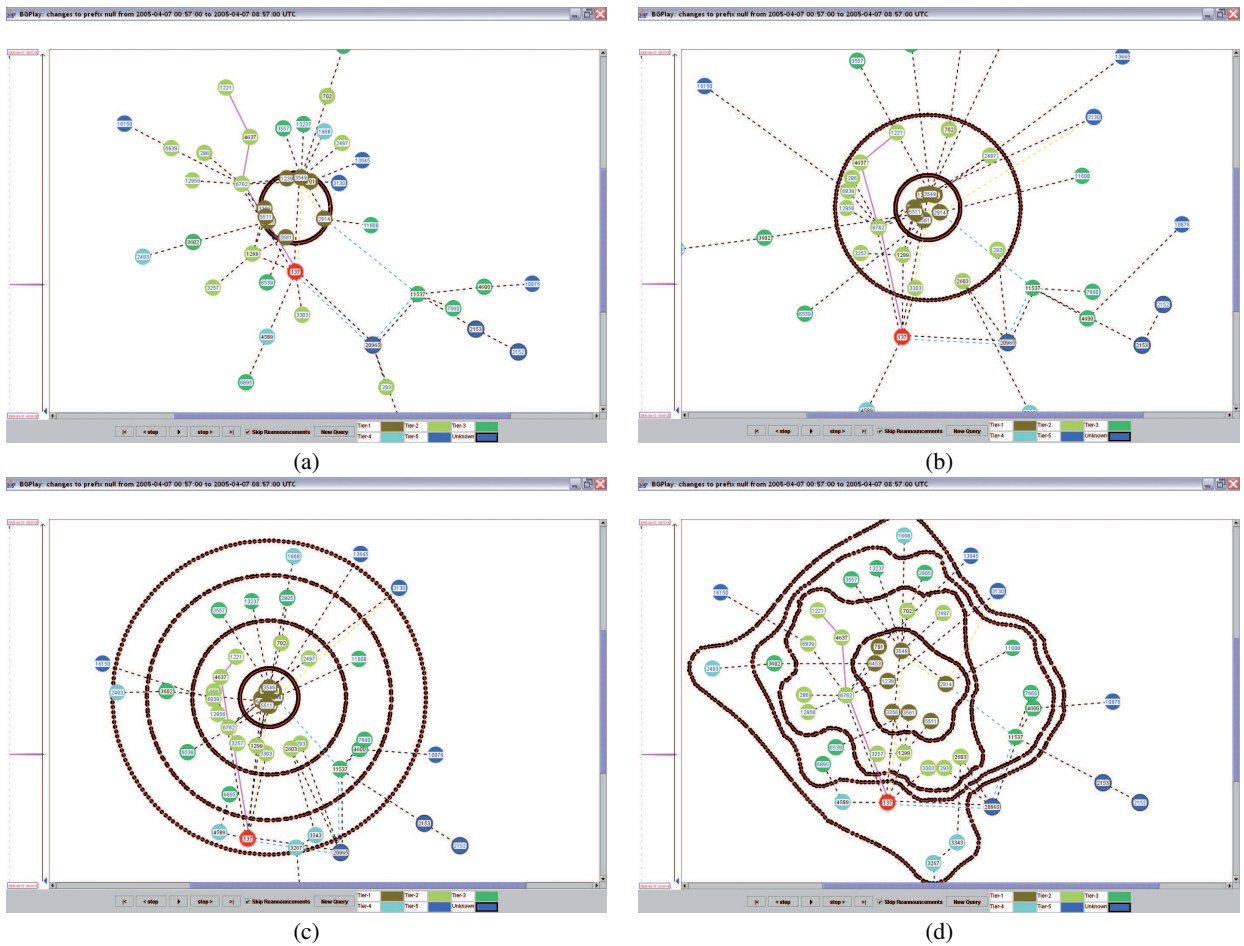


Fig. 4. Example of application of the algorithm. (a) Phase 1, the addition of the first fence. (b) Phase 1, the addition of the second fence. (c) Phase 1, the addition of the fourth fence. (d) Final layout after Phase 2. The actual picture shown to the user is presented in Fig. 2(a).

edges. **Node distance ratio (NDR)**, i.e., the ratio between the distance of the nearest pair of nodes and that of the farthest one. This measures the “resolution rule” of the drawing [18].

Concerning aesthetic criteria that are specific for the application, we selected the following ones. Edge region crossings (ERX), i.e., the number of edges that cross the border of the same region more than once. Coastline indention vector (CIV), that is an index CI_i for each region i measuring how smooth is the border of that region. In fact, we noticed that “nice looking” regions have borders which are not too much indented. We define $CI_i = \frac{b_i}{2\sqrt{\pi a_i}}$, where b_i and a_i are the outer perimeter of border B_i , and the area enclosed by it, respectively. Observe that, $CI_i \geq 1$, where $CI_i = 1$ if B_i is drawn as a circle. Also, it would be reasonable to have a coastline indention that slightly increases with i , thus having $CI_i \leq CI_{i+1}$. Density variation vector (DVV), that measures how the density changes from region to region. The density variation between subsequent areas is defined as $DV_i = \frac{D_i - D_{i+1}}{D_i}$, where D_i is the number of nodes in region i divided by A_i . Essentially, we would like to have drawings that show a small change of their density when moving from a region to an adjacent one.

The number of prefixes currently announced in the Internet is about 200,000 originated by about 20,000 ASes. To perform our experiments we selected a test suite of 141 prefixes in the following way. We selected all the Tier 1 ASes (which are 21). Then, we randomly selected 30 ASes in each tier. For each selected AS we randomly picked up a prefix. For each prefix we applied the algorithm described in Section 4 with four different choices of the rigidity of the fences. We shall refer to such different versions of the algorithm with CIRCLES, RIGID FENCES, SOFT FENCES, and LOOSE FENCES. CIRCLES corresponds

to having fixed fences, with the shape of circles, which is equivalent to set the mobility factor to zero (see Section 4). It constitutes a reference point for the quality of the drawings. SOFT FENCES corresponds to the choice we have done in the current version of the system. LOOSE FENCES have the mobility factor equal to 4/3 the mobility factor used by SOFT FENCES. Finally, RIGID FENCES corresponds to an intermediate choice between CIRCLES and SOFT FENCES and has a mobility factor equal to 1/3 of SOFT FENCES.

All drawings computed with all four versions of the algorithm have ERX=0.

Fig. 5 shows the density of the values of EEX measured in our experiments. Namely, the x-axis reports the values of EEX while the y-axis reports the number of drawings having that value of EEX. Since EEX is a continuous variable, we grouped its values in 15 intervals. For example, the y-value 11 corresponding to the x-value 0.605 on the curve for CIRCLES means that there are 11 drawings with value of EEX between 0.56 and 0.65. We use density charts also for the other quality measures.

As Fig. 5 shows, SOFT FENCES gives values of EEX that are lower, in average, with respect to CIRCLES. A similar behavior, but less marked, we have for RIGID FENCES. LOOSE FENCES shows only a marginal improvement on SOFT FENCES. The average EEX values are 0.189 for LOOSE FENCES, 0.196 for SOFT FENCES, 0.364 for CIRCLES, and 0.24 for RIGID FENCES. This happens because fixed or rigid fences do not allow (or only partially allow) the spring embedder algorithm to disentangle the drawing.

Fig. 6 and 7 shows the density of the values of NDR and ELSD, respectively. Both figures put in evidence the beneficial effect of having fences that are free to move. Again, SOFT FENCES and LOOSE

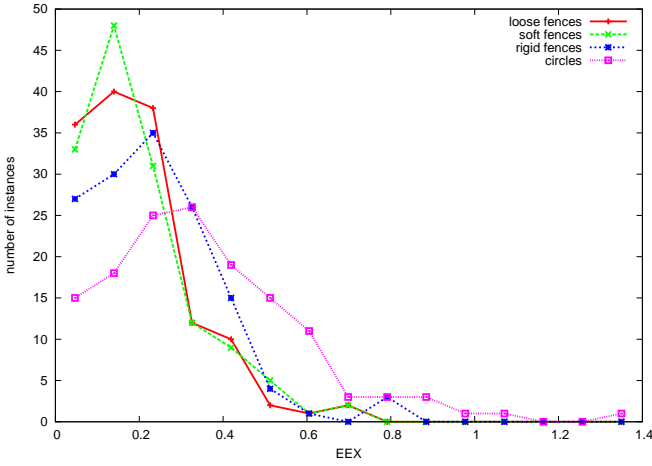


Fig. 5. Density of EEX (normalized number of edge crossings). CIRCLES and RIGID FENCES are more affected by edge crossings.

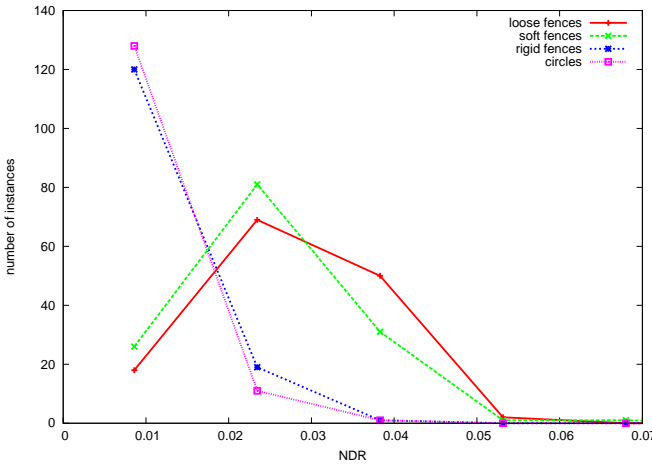


Fig. 6. Density of NDR (node distance ratio). LOOSE FENCES and SOFT FENCES show a better “resolution rule” for their nodes.

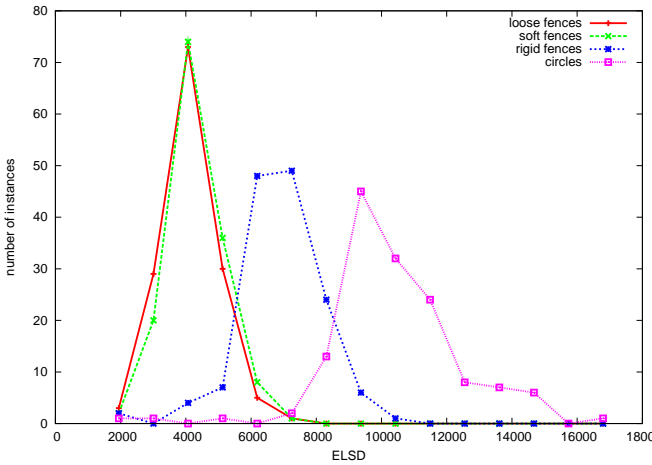


Fig. 7. Density of ELSD (normalized edge-length standard deviation). LOOSE FENCES and SOFT FENCES show more homogeneous edge lengths.

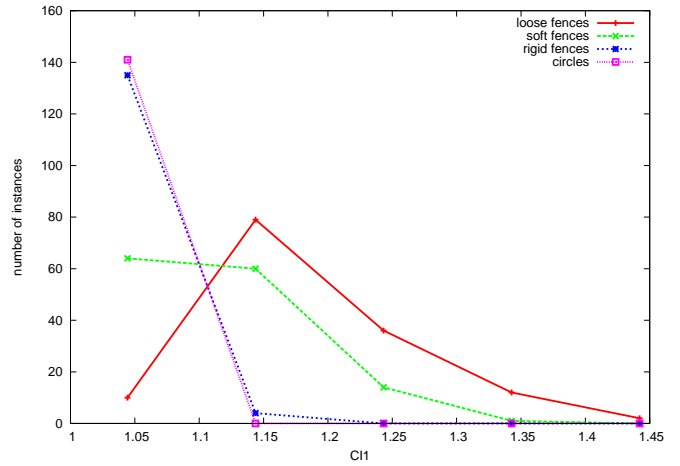


Fig. 8. Density of CI_1 (coastline indentation of region 1). Augmenting the mobility factor progressively increases the indentation of the border of region 1.

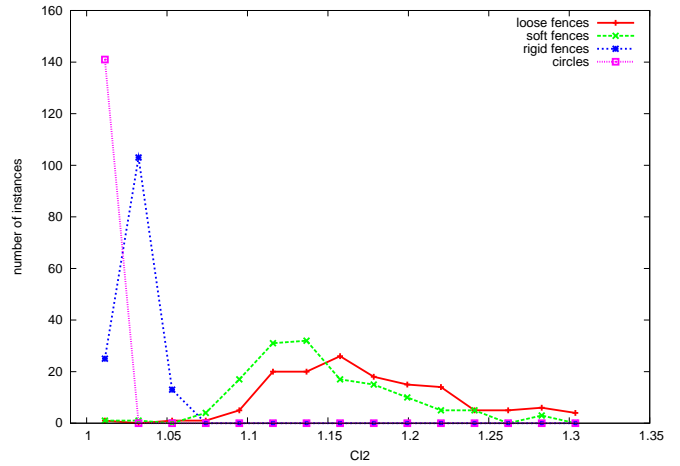


Fig. 9. Density of CI_2 (coastline indentation of region 2). LOOSE FENCES and SOFT FENCES have a comparable quality with respect of this measure. CIRCLES and RIGID FENCES produce, obviously, smoother borders.

FENCES show a very similar behavior.

Figs. 8 and 9 show the density of CI_1 and CI_2 . The plots for CI_3 and CI_4 are similar to that of CI_2 . The plots put in evidence that the satisfactory values of EEX, NDR, and ELSD are obtained at the expense of larger contour indentation. The worst cases are the ones of CI_1 and CI_2 . This is somehow counter-intuitive since we could expect higher indentation for the most external regions. However, such regions have typically less nodes. Hence, the contours tend to stretch. Also, for these measures, it is apparent that LOOSE FENCES performs worse than SOFT FENCES.

Fig. 10 shows that SOFT FENCES and LOOSE FENCES perform much better than the other settings in terms of the density variation between subsequent areas. This is another positive effect of having fences that can move. For brevity we omit results for DV_2 and DV_3 that are quite similar.

Overall, we think that SOFT FENCES is a good trade-off between the quality of the borders and the quality of the graph layout. LOOSE FENCES offers a small improvement for EEX, NDR, ELSD, and DVV, while performs worse for CIV, especially for CI_1 , CI_3 , and CI_4 .

We think that the limitations of our approach are basically the same of conventional spring embedders, which are known to perform poorly for graphs with high density (ratio between the number of edges and the number of nodes). However, in our application, the density is quite

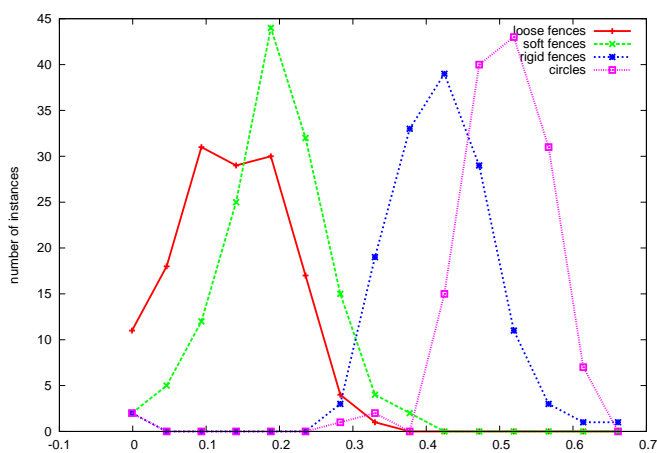


Fig. 10. Density of DV_1 (density variation between areas A_1 and A_2). A more homogeneous density is shown by LOOSE FENCES and SOFT FENCES.



Fig. 11. The map of the AS-graph with highest density in the test suite (Algorithm SOFT FENCES).

low. In our experiments density ranges from 0.98 to 1.16 (average is 1.04). The drawing of the AS-graph with the highest density is shown in Fig. 11.

Three network topographic maps of the same AS-graph produced for the evaluation are in Fig. 12.

We have done some evaluating interviews with ISP network managers to get subjective evaluative remarks. Namely, we proposed our prototype to people working with NaMeX [4], the second Internet Exchange Point in Italy, grouping 22 large and medium size ISPs. We asked for comments and feedbacks on the usability of the system, also compared with the current version of BGPlay. They found interesting and useful to have the relevance of each AS highlighted. The choice of the topographic approach has been appreciated, as well as the quality of the drawings. We collected feedback on minor usability issues related to interaction that were easy to amend

6 Conclusions

BGPlay [16] is a popular service used by network operators to visualize how an IP prefix propagates through the Internet. This paper starts from the observation that for an ISP it would be interesting to enrich

the visual information already provided by BGPlay with information on the hierarchy of the ISPs. For example, for an ISP that is on a low level of the hierarchy it would be interesting to know if, for reaching the rest of the Internet its prefixes have to “climb the hill” passing through tier-one players or if it is sufficient, at least for some of them, to follow paths that are composed by lower level providers. This can be useful both for operational purposes and for studying the impact in terms of routing of the peering choices of an operator.

To do that, we introduced a new visualization metaphor, and presented an algorithm to produce layouts consistent to such metaphor. Our approach is to represent the graph of the ASes and the Internet Hierarchy as a topographic map, where the high level ASes, i.e. the Internet backbone, are placed at the top of the mountain.

This kind of metaphor permits an effective visualization of the routing paths that highlights the “jumps” they perform from a tier to another of the hierarchy. Further, the topographic map represents a special case of clustered drawing and we believe that the techniques proposed in this paper might be successfully applied in different applicative contexts as, for example, to represent social networks on a hierarchically organized community or procedure calls in the presence of layered software architectures.

As future lines of research we suggest the following: combine the current information about Internet routing with information on the amount of traffic flows from and to the focal ISP; integrate the inter-domain routing information currently provided by BGPlay with intra-domain routing; compute AS hierarchies with algorithms alternative to [30].

Acknowledgements

We wish to thank Gabriele Barbagallo and Luca Salvi of the NaMeX [4] Internet Exchange Point for the time they dedicated to our prototypical tool and for their useful suggestions. This work was supported in part by European Commission - Fet Open project DELIS - Dynamically Evolving Large Scale Information Systems - Contract no 001907 and by “Project ALGO-NEXT: Algorithms for the Next Generation Internet and Web: Methodologies, Design, and Experiments”.

References

- [1] The atlas of cyberspace. <http://www.cybergeography.org/atlas/>.
- [2] The internet mapping project. <http://research.lumeta.com/ches/map/index.html>.
- [3] Ip mapping. <http://www.fractalus.com/steve/stuff/ipmap/>.
- [4] NaMeX, Nautilus Mediterranean Exchange. <http://www.namex.it>.
- [5] Routing Information Service of the RIPE (RIS). <http://www.ripe.net/ripenncc/pub-services/np/ris/>.
- [6] University of Oregon RouteViews project. <http://www.routeviews.org>.
- [7] Walrus - graph visualization tool. <http://www.caida.org/tools/visualization/walrus/>.
- [8] Wikipedia - topographic map. http://en.wikipedia.org/wiki/Topographic_map.
- [9] C. Bachmaier, F. Fischer, and M. Forster. Radial coordinate assignment for level graphs. In *Proc. COCOON 2005*, volume 3595 of *LNCS*, pages 401–410. Springer, 2005.
- [10] G. Di Battista, M. Patrignani, and M. Pizzonia. Computing the types of the relationships between autonomous systems. In *Proc. IEEE INFOCOM*, volume 1, pages 156–165, 2003.
- [11] F. J. Brandenburg, M. Himsolt, and C. Rohrer. An experimental comparison of force-directed and randomized graph drawing algorithms. In *Graph Drawing (Proc. GD'95)*, volume 1027 of *LNCS*, pages 76–87, 1996.
- [12] U. Brandes and S.R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization*, 2(1):40–50, 2003.
- [13] U. Brandes, P. Kenis, and D. Wagner. Communicating centrality in policy network drawings. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):241–253, 2003.

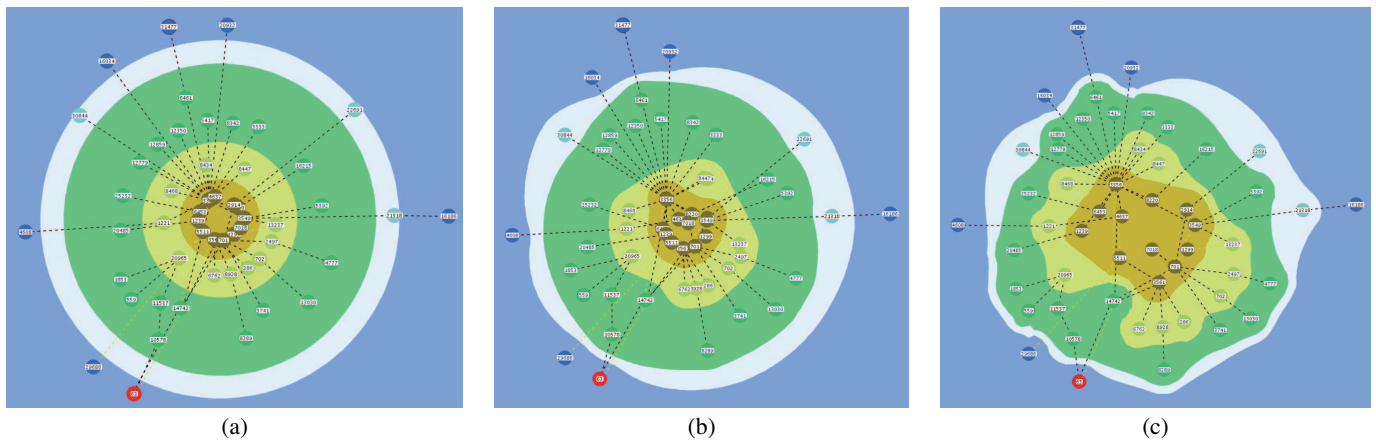


Fig. 12. Three topographic maps of the same AS-graph. They were produced by different algorithms: (a) CIRCLES, (b) RIGID FENCES, and (c) SOFT FENCES.

- [14] A. Carmignani, G. Di Battista, W. Didimo, F. Matera, and M. Pizzonia. Visualization of the high level structure of the internet with hermes. *J. Graph Alg. Appl.*, 6(3):281–311, 2002.
- [15] M. J. Carpano. Automatic display of hierarchized graphs for computer aided decision analysis. *IEEE Trans. Syst. Man Cybern.*, SMC-10(11):705–715, 1980.
- [16] L. Colitti, G. Di Battista, F. Mariani, M. Patrignani, and M. Pizzonia. Visualizing interdomain routing with BGPlay. *J. Graph Alg. Appl.*, 9(1):117–148, 2005.
- [17] K.C. Cox, S.G. Eick, and T. He. 3D geographic network displays. *SIGMOD Record*, 25(4):50–54, 1996.
- [18] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.
- [19] P. Eades and M.L. Huang. Navigating clustered graphs using force-directed methods. *J. Graph Alg. Appl.*, 4(3):157–181, 2000.
- [20] Q. W. Feng, R. F. Cohen, and P. Eades. How to draw a planar clustered graph. In *Proc. COCOON'95*, volume 959 of *LNCS*, pages 21–30. Springer-Verlag, 1995.
- [21] Y. Frishman and A. Tal. Dynamic drawing of clustered graphs. In *INFOVIS '04*, pages 191–198. IEEE Computer Society, 2004.
- [22] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Softw. Pract. Exper.*, 21(11):1129–1164, 1991.
- [23] L. Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Trans. on Networking*, 9(6):733–745, 2001.
- [24] B. Huffaker, D. Plummer, D. Moore, and k claffy. Topology discovery by active probing. In *Symposium on Applications and the Internet (SAINT)*, 2002. <http://www.caida.org/outreach/papers/2002/SkitterOverview/>.
- [25] D. Massey M. Lad and L. Zhang. Visualizing internet routing dynamics using link-rank. Technical Report TR050010, UCLA, 2005.
- [26] H. C. Purchase. Effective information visualisation: A study of graph drawing aesthetics and algorithms. *Interact. Comput.*, 13(2):147–162, 2000.
- [27] H. C. Purchase. Metrics for graph drawing aesthetics. *J. of Visual Lang. and Comput.*, 13(5):501–516, 2002.
- [28] J. M. Six and I. G. Tollis. Automated visualization of process diagrams. In *Proc. Graph Drawing (GD '01)*, pages 45–59, 2001.
- [29] J.W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, Reading, MA, 1999.
- [30] L. Subramanian, S. Agarwal, J. Rexford, and R.H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *Proc. IEEE INFOCOM*, 2002.
- [31] K. Sugiyama and K. Misue. Graph drawing by the magnetic spring model. *J. Visual Lang. Comput.*, 6(3):217–231, 1995.
- [32] K. Sugiyama and K. Misue. A simple and unified method for drawing graphs: Magnetic-spring algorithm. In *Graph Drawing (Proc. GD '94)*, volume 894 of *LNCS*, pages 364–375, 1995.
- [33] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical systems. *IEEE Trans. Syst. Man Cybern.*, SMC-11(2):109–125, 1981.
- [34] A. S. Tanenbaum. *Computer Networks*. Prentice-hall Int., 1996.
- [35] V. E. Waddle. Graph layout for displaying data structures. In *Graph Drawing (Prof. GD '00)*, pages 241–252. Springer-Verlag, 2001.
- [36] C. Walshaw. A multilevel algorithm for force-directed graph drawing. In *Graph Drawing (Proc. GD '00)*, pages 171–182, 2001.
- [37] K. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *Proc. INFOVIS '01*, pages 43–50. IEEE Computer Society, 2001.